

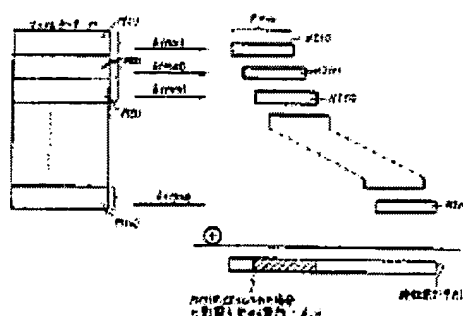
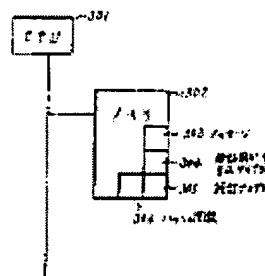
# VERIFICATION DATA GENERATING SYSTEM

**Patent number:** JP3151738  
**Publication date:** 1991-06-27  
**Inventor:** FUKUZAWA YASUKO; others: 02  
**Applicant:** HITACHI LTD  
**Classification:**  
 - International: H04L9/06; H04L9/14  
 - european:  
**Application number:** JP19890288887 19891108  
**Priority number(s):**

## Abstract of JP3151738

**PURPOSE:** To detect the presence of forgery of file content and the forged part by splitting a file data, applying logical operation while deviating one by one bit compressed sentence generated to each of split file data.

**CONSTITUTION:** A message 303 for an object of verification, a verification data generating program 304, a verification program 305 and a hash function 306 are stored in a memory 302 in a computer and a CPU 301 uses the data to generate and verify a verification data. A message M being an object for generating the verification data is decided into (n) as  $M(i)$  ( $i=0 \dots n$ ), and a partial compression sentence  $HI(i)(p\text{-bit})$  is generated with respect to the  $M(i)$  by using the hash function (h). The generated partial compression sentences  $HI(i)$  are deviated one by one bit to obtain exclusive OR and the result is used for the verification data  $HI$  in  $(p+n-1)$  bits. That is, the exclusive OR between the 2nd bit data of the  $HI(1)$  and the 1st bit data of the  $HI(2)$  is the 2nd bit data of the verification data.



Data supplied from the esp@cenet database - Worldwide

BEST AVAILABLE COPY

⑩ 日本国特許庁(JP)

訂正有り  
⑪ 特許出願公開

⑫ 公開特許公報(A)

平3-151738

⑬ Int. Cl.<sup>5</sup>

H 04 L 9/06  
9/14

識別記号

庁内整理番号

⑭ 公開 平成3年(1991)6月27日

6914-5K H 04 L 9/02

Z

審査請求 未請求 請求項の数 6 (全12頁)

⑮ 発明の名称 検証用データ生成方式

⑯ 特 願 平1-288887

⑰ 出 願 平1(1989)11月8日

⑱ 発 明 者 福 澤 享 子 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内  
⑲ 発 明 者 宝 木 和 夫 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内  
⑳ 発 明 者 佐 々 木 良 一 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内  
㉑ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地  
㉒ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

検証用データ生成方式

2. 特許請求の範囲

1. 電子的なメッセージMの検証用データ生成方式において、

該メッセージMをn個に分割し、 $M = M(1) || M(2) || \dots || M(i) || \dots || M(n)$ とし、

分割したn個の該メッセージ $M(i)$  ( $i = 1, 2, \dots, n$ ) に対し、ハッシュ関数hによつてpビットの圧縮文 $H(i)$  ( $H(i) = h(M(i))$ ) ( $i = 1, 2, \dots, n$ ) を作成し、

該圧縮文 $H(i)$  ( $i = 1, 2, \dots, n$ ) をmビット ( $1 \leq m \leq p$ ) ずつずらして論理演算を施した  $(p + m(n-1))$  ビットのデータを上記メッセージMと対応する検証用データHとすることを特徴とする検証用データ作成方式。

2. 電子的なメッセージMの検証用データ生成方式において、

該メッセージMをn個に分割し、 $M = M(1)$

$|| M(2) || \dots || M(i) || \dots || M(n)$ とし、

分割したn個の該メッセージ $M(i)$  ( $i = 1, 2, \dots, n$ ) に対し、ハッシュ関数hによつてpビットの圧縮文 $H(i)$  ( $H(i) = h(M(i))$ ) ( $i = 1, 2, \dots, n$ ) を作成し、

該 $H(i)$  ( $i = 1, 2, \dots, n$ ) の左半分を $H(i)L$  ( $i = 1, 2, \dots, n$ )、右半分を $H(i)R$  ( $i = 1, 2, \dots, n$ ) とし、

$H'(i)L = H(j)L$  ( $1 \leq j \leq n$  であり、 $H'(j)L \neq H'(p)L$  ( $p < i$ )) となるように $H(i)L$  を再配置した $H'(i)L$  ( $i = 1, 2, \dots, n$ ) を生成し、

$H'(i)R = H(j)R$  は、 $1 \leq j \leq n$  であり、 $H'(j)R \neq H'(p)R$  ( $p < i$ ) であり、

$H'(f)R$  ( $i-2 \leq f < i$ ,  $i < f \leq i+2$ )  $\neq H(k)R$  ( $j-2 \leq k < j$ ,  $j < k \leq j+2$ ) であり、

$H'(m)R$  ( $j-5 \leq m \leq j-1$ )  $\neq H(n)R$  ( $i+1 \leq n \leq i+5$ ) となるように $H(i)R$  を再配置した $H'(i)R$  ( $i = 1, 2, \dots, n$ ) を生

(1)

(2)

(2)

成し、

該再配列した結果を $H'(i)$  ( $i=1, 2 \dots n$ )とし、

該 $H'(i)$  ( $i=1, 2 \dots n$ )を $m$ ビットずつずらして論理演算を施した( $p+m(n-1)$ )ビットのデータを上記メッセージ $M$ と対応する検証用データ $H'$ とすることを特徴とする検証用データ作成方式。

3. 請求項2に記載の検証用データ生成方式において、

作成した $p$ ビットの部位圧縮文 $H(i)$  ( $i=1, 2 \dots n$ )の各々を、

$s$  ( $p$ の公約数であり、 $s \neq 1, 2, p$ )個で分割し、 $H(i)(1), H(i)(2), \dots, H(i)(s)$  ( $i=1, 2 \dots n$ )とし、

$H'(i)(1)=H(j)(1)$  ( $1 \leq j \leq n$ であり、 $H'(j)(1) \neq H'(p)(1)$  ( $p < i$ ))となるように $H(i)(1)$ を再配置した $H'(i)(1)$  ( $i=1, 2 \dots n$ )を生成し、

(3)

ータとすることを特徴とする電子捺印方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、電子化されたファイルの検証用データ生成方式に関する。

〔従来の技術〕

電子データの活用や保存が盛んになるに従い、ファイルの正当性を認証する技術はますます重要になる。ファイルの内容を圧縮文によつて表す技術は、ファイル内容を確認する有効な技術の1つである。これは大量のファイルデータをハッシュ関数によつて圧縮し、64ビット長程度の認証子(圧縮文)を作成する技術である。ファイルデータの内容が1ビットでも変わると、全く異なった圧縮文を生成してしまうため、データ改ざんの有無を検知することができる。この技術に、公開鍵暗号を応用し、上記記載の電子取引の安全性を確保する電子捺印が提案されている。

〔発明が解決しようとする課題〕

しかし、改ざんの有無を検知することができて

(5)

$1 < r \leq s$ の $r$ について、

( $1 \leq x \leq n$ )であり、 $H'(i)(r)$ が影響を及ぼす $p$ ビットの範囲に存在する $x$ に関し、

$H'(i)(r) \neq H(x)(r)$ でないように $H(i)(r)$ を再配置した $H(i)'(r)$  ( $i=1, 2 \dots n$ )を生成し、

該再配列した結果を $H'(i)$  ( $i=1, 2 \dots n$ )とする再配列方式。

4. 請求項2もしくは3に記載の検証用データ生成方式において、

$n$ 個に分割したメッセージ

$n > (2p/s - 1)^2$ の関係式が成立する検証用データ生成方式。

5. ファイル $M$ の圧縮文に、請求項1乃至4のいずれかに記載の、検証用データを付加することとを特徴とするファイル認証方式。

6. 請求項1乃至5のいずれかに記載のファイルの圧縮文に、時刻等の状況データを付加したデータを、公開鍵暗号の秘密鍵を用いて、公開鍵暗号で暗号処理し、これを該ファイルの認証デ

(4)

も、改ざんの箇所を検知することはできない。

この問題に対処するため、改良案を考案していた(特願昭62-321220号)。これはファイルを階層化し、階層化した個々のファイルデータに対して圧縮文を生成、保存することで、後日の改ざんを検知する。しかし、この方法だと、個々の圧縮文を保存するため、保存すべき情報量が多くなるという不満足な点があった。

〔課題を解決するための手段〕

上記問題に対し、検証用データ生成方式を考案した。これは、ファイルデータを分割し、分割した個々のファイルデータに対して作成した圧縮文を1ビットずつずらして論理演算を行う。あるいは、検知確率を向上させるために、個々の圧縮文を2つに分割し、最適に配置し、配置し直した各圧縮文に対し、論理演算を行う。

〔作用〕

前記技術的手段により、次の効果が生じる。

1. ファイルの検証用データ(例えば、416ビット)生成後、ファイルデータが改ざんされた場

(6)

合、 $\left(1 - \frac{1}{2^{416}}\right)$  の確率で改ざんの有無を検知することができる。

2. ファイル改ざん前後の検証用データの相違によつて、ファイルデータの改ざん位相をかなりの確率で検知することが可能になる。

〔実施例〕

第1図～第9図において、本発明の実施例を示す。

〔実施例1〕

第1図～第4図において、電子的なメッセージMの検証用データ生成方式、およびメッセージ改ざん検証の一例を示す。

第1図は、検証用データ生成の一方法を示すフロー図である。第2図は、メッセージの改ざんを検知する一方法を示すフロー図である。

第3図は、処理を行う計算機の一例である。

第4図は、第1図の検証用データ生成の実際のイメージを示す。

(7)

step 1 0 6 : i に 1 を加え、step 1 0 2 に進む。  
step 1 0 7 : カウント i を 0 に設定する。  
step 1 0 8 : カウント i が、 $i < n$  ならば step 1 0 9 に進む、 $i \geq n$  ならば step 1 1 1 に進む。  
step 1 0 9 : 作成した部位圧縮文  $H I(i)$  を 1 ビットずつずらして排他的論理和を求め、これを  $(p+n-1)$  ビットの検証用データ  $H I$  とする。つまり、 $H I(1)$  の 2 ビット目と  $H I(2)$  の 1 ビット目の排他的論理和が検証用データの 2 ビット目となる。

$H I(1)$  の 3 ビット目と  $H I(2)$  の 2 ビット目と、

$H I(3)$  の 1 ビット目の排他的論理和が検証用データの 3 ビット目となる。

step 1 1 0 : i に 1 を加え、step 1 0 8 に進む。

step 1 1 1 : 検証用データ  $H I$  を出力する。

step 1 1 2 : 終わり。

次に、上記の検証用データ  $H I$  作成時のメッセージMと現時点でのメッセージM' が同等であることを検証する例を第2図のフローに従って示す。

(8)

(3)

第3図において、計算機上のメモリ302に、検証対象のメッセージ303、検証用データ生成プログラム304、検証プログラム305、およびハッシュ関数306が蓄えられており、これらを用いてCPU301によつて検証用データ生成と検証を行う。検証用データ生成手段を第1図のフローのステップ(step)に従って示す。

step 1 0 0 : 始め

step 1 0 1 : 検証用データ生成の対象となるメッセージの名称Mを設定し、メッセージをn個に分割し、個々を $M(i)$  ( $i = 0 \dots n$ ) とする。また、カウントiを0に設定する。

step 1 0 2 : カウントiが、 $i < n$  ならば step 1 0 3 に進む、 $i \geq n$  ならば step 1 0 7 に進む。

step 1 0 3 : メッセージ $M(i)$ を読み込む。

step 1 0 4 :  $M(i)$  に対して、部位圧縮文  $H I(i)$  (p ビット) をハッシュ関数hを用いて生成する。

step 1 0 5 : 部位圧縮文  $H I(i)$  をメモリ302上に退避する。

(8)

step 2 0 0 : 始め

step 2 0 1 : 既に生成済みのMの検証用データ  $H I$  を入力する。

step 2 0 2 : 検証の対象であるメッセージM' について、step 1 0 0 から step 1 1 2 に従い、検証用データを生成し、これを  $H I'$  とする。

step 2 0 3 : 検証用データ  $H I$  と  $H I'$  を比較し、不一致部分を検知する。一致した場合は step 2 0 4 に進む、不一致の場合には step 2 0 5 に進む。

step 2 0 4 : メッセージMとM' は同一であると判定し、step 2 0 7 に進む。

step 2 0 5 : メッセージMとM' は同一でないと判定する。

step 2 0 6 : 検知した不一致部分位置から、メッセージM' の改ざん部位を検知する。

例えば、第4図において、 $H I$  と  $H I'$  を比較するとd—Hの位置が影響を受けていた場合、 $H I(3)$  と  $H I'(3)$  が一致しなかったことが自明であり、この結果M(3)が改ざんされたことが

(10)

(4)

わかる。

step 207: 終わり。

〔実施例 2〕

第 5 図～第 8 図において、電子的なメッセージ M の検証用データ生成方式、およびデータ改ざんを検証する他の例を示す。

第 5 図は、検証用データ生成の一方法を示すフロー図である。第 6 図は、メッセージの改ざんを検知する一方法を示すフロー図である。

第 7 図、第 8 図は、検証用データ生成の実際のイメージを示めす。

第 5 図、および第 7 図、第 8 図において、検証用データ生成の手順を示す。

メッセージ M を n 個に分割し、各分割メッセージに対して p ビットの部分圧縮文を生成し、部分圧縮文を s 個に分割し、これを再配置して検証用データの生成を行う。この時、再配置における分散を高めるために、例えば n, s, p は次の関係式が成り立つようにする。

$$n > (2p/s - 1)^2$$

(11)

step 508: カウント i が、 $i < 26$  ならば step 509 に進み、 $i \geq 26$  ならば step 512 に進む。

step 509: 作成した部位圧縮文  $H_{II}(i)$  ( $i = 1, 2, \dots, 26$ ) の左側 3 ビットを  $H_{II}(i)L$ 、右側 3 ビットを  $H_{II}(i)R$  とする。

$H_{II}'(i)L = H_{II}(i)L$  ( $i = 1, 2, \dots, 26$ ) とする。

$H_{II}'(i)R = H_{II}(j)R$  ( $i = 1, 2, \dots, 26$ ) とし、j を次のルールに従い再配置する。

- (1)  $1 \leq j \leq 26$  であり、
- (2)  $H_{II}(j)R \neq H_{II}'(p)R$  ( $p < j$ ) であり、
- (3)  $H_{II}'(i)R$  が影響を与える  $H_{II}'(k)R$  ( $i - 2 \leq k < i$ ,  $i < k \leq i + 2$ ) は、  
 $H_{II}'(j)L$  が影響を与える  $H_{II}(f)R$  ( $j - 2 \leq f < j$ ,  $j < f \leq j + 2$ ) でなく、
- (4)  $H_{II}(j)L$  が影響を与える  $H_{II}'(m)R$  ( $j - 5 \leq m \leq j - 1$ ) には、 $H_{II}'(i)L$  が影響を与える  $H_{II}(n)R$  ( $i + 1 \leq n \leq i + 4$ ) ではない。

(13)

ここでは、ファイル M を 26 個に分割し、作成する各部位圧縮文は 6 ビットとし、各部位圧縮文は 2 つに分割して再配置する。各部位圧縮文より生成する検証用データ 31 ビットとする。

step 500: 始め

step 501: 検証用データ生成の対象となるメッセージの各称 M を設定し、メッセージを 26 個に分割し、個々を  $M(i)$  ( $i = 1, 2, \dots, 26$ ) とする。また、カウント i を 0 に設定する。

step 502: カウント i が、 $i < 26$  ならば step 503 に進み、 $i \geq 26$  ならば step 507 に進む。

step 503:  $M(i)$  を読み込む。

step 504:  $M(i)$  に対して、ハッシュ関数 h を用いて部位圧縮文  $H_{II}(i)$  (6 ビット) を作成する。

step 505: 部位圧縮文  $H_{II}(i)$  をメモリ 302 上に退避する。

step 506: i に 1 を加え、step 501 に進む。

step 507: カウント i を 0 に設定する。

(12)

step 510: 作成した部位圧縮文  $H_{II}'(i)$  を 1 ビットずつずらして排他的論理和を求め、これを  $(p + n - 1)$  ビットの検証用データ  $H_{II}$  とする。つまり、 $H_{II}'(1)$  の 2 ビット目と  $H_{II}'(2)$  の 1 ビット目の排他的論理和が検証用データの 2 ビット目となる。 $H_{II}'(1)$  の 3 ビット目と  $H_{II}'(2)$  の 2 ビット目と、 $H_{II}'(3)$  の 1 ビット目の排他的論理和が検証用データの 3 ビット目となる。

step 511: i に 1 を加え、step 508 に進む。

step 512: 検証用データ  $H_{II}$  を出力する。

step 513: 終わり。

上記手順に従い生成した検証用データの例が第 7 図である。

次に、上記の検証用データ  $H_{II}$  作成時のメッセージ M と現時点でのメッセージ M' が同等であることを検証する例を第 8 図のフローに従って示す。

step 600: 始め

step 601: 既に生成済みの M の検証用データ  $H_{II}$  を入力する。

(14)

(5)

step 6 0 2 : 検証の対象であるメッセージ  $M'$  について、step 5 0 0 から step 5 1 3 に従って、検証用データ  $H \Pi$  生成と同じ型の再配列を行い、メッセージ  $M'$  の検証用データを生成し、これを  $H \Pi'$  とする。

step 6 0 3 : 検証用データ  $H \Pi$  と  $H \Pi'$  を比較し、一致した場合は step 6 0 4 に進み、不一致の場合には step 6 0 5 に進む。

step 6 0 4 : メッセージ  $M$  と  $M'$  は同一であると判定し、step 6 0 7 に進む。

step 6 0 5 : メッセージ  $M$  と  $M'$  は同一でないと判定される。

step 6 0 6 : ファイルデータ改ざん前後の改ざん検知用圧縮文  $H$  と  $H \Pi'$  の比較する。 $M(5)$  が改ざんされた場合には、検証用データ  $H \Pi'$  において、 $D 1$ 、および  $D 2$  の部分で一致しない。

従って、改ざん部位の構成より、次のように判断できる。

改ざん部位 =  $D 1 \cap D 2$

$= (H \Pi(3) L U H \Pi(4) L U H \Pi(5) L U H \Pi(6) L U H \Pi(7)) L$   
(15)

持つ検証用データが多くなり、一方、改ざん位置の検知確率は向上する。

〔変形例 3〕

実施例 2 において、分割した部位の各圧縮文を、3 以上に複数に分割する。例えば、 $H \Pi$  を 3 分割し  $H \Pi L(i)$ ,  $H \Pi M(i)$ ,  $H \Pi R(i)$  ( $i = 1, 2 \dots n$ ) とし、

$H \Pi'(i) L = H \Pi(i) L$  ( $i = 1, 2 \dots 26$ ) とする。

$H \Pi'(i) M = H \Pi(j) R$  ( $i = 1, 2 \dots 26$ ) とし、 $j$  を次のルールに従い再配置する。

- (1)  $1 \leq j \leq 26$  であり、
- (2)  $1 \leq k \leq 26$  であり、 $H \Pi'(j) M$  が影響を与える範囲に存在する  $k$  に関して、 $H \Pi'(j) M \neq H \Pi'(k) M$  とする。

また、 $H \Pi'(i) R = H \Pi(j) R$  ( $i = 1, 2 \dots 26$ ) とし、 $j$  を次のルールに従い再配置する。

- (1)  $1 \leq j \leq 26$  であり、
- (2)  $1 \leq k \leq 26$  であり、 $H \Pi'(j) R$  が影響を

(17)

$U(H \Pi(7) R U H \Pi(10) R U H \Pi(13) R U H \Pi(16) R)$

$\cap$

$(H \Pi(17) L U H \Pi(18) L U H \Pi(19) L U H \Pi(20) L U H \Pi(21)) L U (H \Pi(23) R U H \Pi(26) R U H \Pi(5) R U H \Pi(1) R U H \Pi(11) R)$

$= H(5)$

$M(5)$  が改ざんされたことが検知できる。

ただし、ここでの  $\cap$  は、論理積であり、 $L$  と  $R$  が対となっていることを意味する。

step 6 0 7 : 終り。

〔変形例 1〕

実施例 1、実施例 2 の検証用データ生成において、生成した各部位圧縮文を、排他的論理和以外の論理演算（論理和、論理積等）によつて処理しても同等の機能を実現することができる。

〔変形例 2〕

実施例 1、実施例 2 の検証用データ生成において、生成した各部位の圧縮文を  $m$  ( $1 \leq m \leq p$ ) ビットずつずらして論理演算処理を行つても同等の機能を実現することができる ( $m$  が多いほど保  
(16)

与える範囲に存在する  $k$  に関して、 $H \Pi'(j) R \neq H \Pi'(k) R$  とする。

再配置した部位圧縮文  $H \Pi'(i)$  を 1 ビットずつずらして排他的論理和を求め、これを  $(p + n - 1)$  ビットの検証用データ  $H \Pi$  とすることも可能である。

〔変形例 4〕

実施例 1、実施例 2 で生成の検証用データ生成方式は、電子取引認証における電子捺印に利用することができる。

step 9 1 1 : 取引伝票 9 0 0 を 3 5 3 の部位に分割し、各部位の圧縮文 (6 4 ビット) を作成し、改ざん部位検知用圧縮文 9 0 3 (4 1 6 ビット) を、実施例 1、あるいは 2 によつて作成する。

step 9 1 2 : 取引伝票 9 0 0 の圧縮文 9 0 2 ( $h(M)$ ) を作成する。

step 9 1 3 : (圧縮文 9 0 2 || 改ざん部位検知用圧縮文 9 0 3 || 3 2 ビットの時刻等の状況データ 9 0 4) を電子捺原文 (5 1 2 ビット) 9 0 1 とし、公開鍵暗号で暗号処理する。

(18)

(6)

## 〔変形例5〕

実施例1、実施例2で生成の検証用データ生成方式は、ファイル認証における認証子として利用することができる。

## 〔変形例6〕

検証用データの生成、および検証をICカード上で実施することも可能である。

## 〔変形例7〕

生成した検証用データを、ICカードに保存することも可能である。

## 〔変形例8〕

実施例2において、検証用データを用いて検証を行う場合に、確率的評価を加えることが可能である。実施例2では、メッセージM(5)の改ざんに伴い、D1、およびD2に影響が生じているが、

D1に最も影響を与える確率が高いのはHLL(5)、HHR(10)であり、

D2に最も影響を与える確率が高いのはHLL(19)、HHR(5)であることから、

$$\text{改ざん部位} = D1 \cap D2$$

(19)

確率で検知することが可能になる。

## 4. 図面の簡単な説明

各図は本発明の実施例を示し、第1図は検証用データ生成の一方法を示すフロー図、第2図は、メッセージの改ざんを検知する方法を示すフロー図、第3図は、処理を行う計算機の一例を示すブロック図、第4図は、第1図の検証用データ生成の実際のイメージを示す説明図、第5図は、検証用データ生成の他の方法を示すフロー図、第6図は、メッセージの改ざんを検知する方法を示すフロー図、第7図と第8図は、第5図の検証用データ生成の実際のイメージを示す説明図、第9図は、検証用データ生成方式を電子捺印に適用した実施例の説明図である。

代理人 弁理士 小川勝男



$$= (H \cap (5) L U H H (10) R)$$

$$\cap$$

$$(H \cap (19) L U H H (5) R)$$

$$= H(5)$$

と検証することができる。

複数箇所の改ざん場所検知等の適用に有効である。

## 〔効果〕

本発明において、ファイル分割情報があり、かつファイル改ざん前後のファイル圧縮文、および改ざん検知用圧縮文が生成できる場合、次のような効果が得られる。

## 1. 改ざん前のファイルの改ざん検知用圧縮文

(ex. 416ビット)生成後、ファイルデータが改ざんされた場合、

$$\left(1 - \frac{1}{2^{416}}\right) \text{ の確率で改ざんの有無を検知す}$$

ることが可能になる。

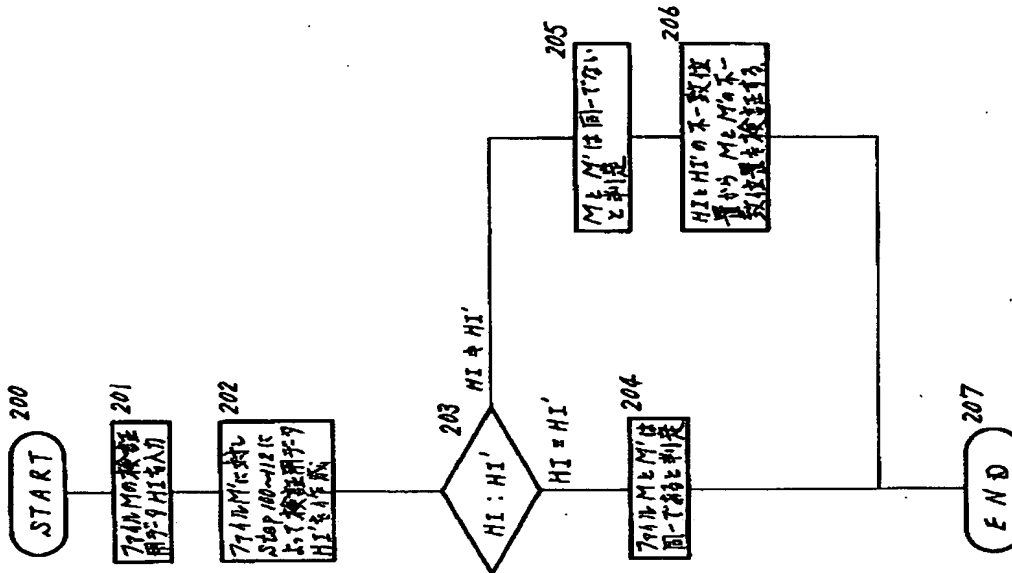
## 2. ファイル改ざん前後の改ざん検証用圧縮文により、ファイルデータの改ざん位置をかなりの

(20)

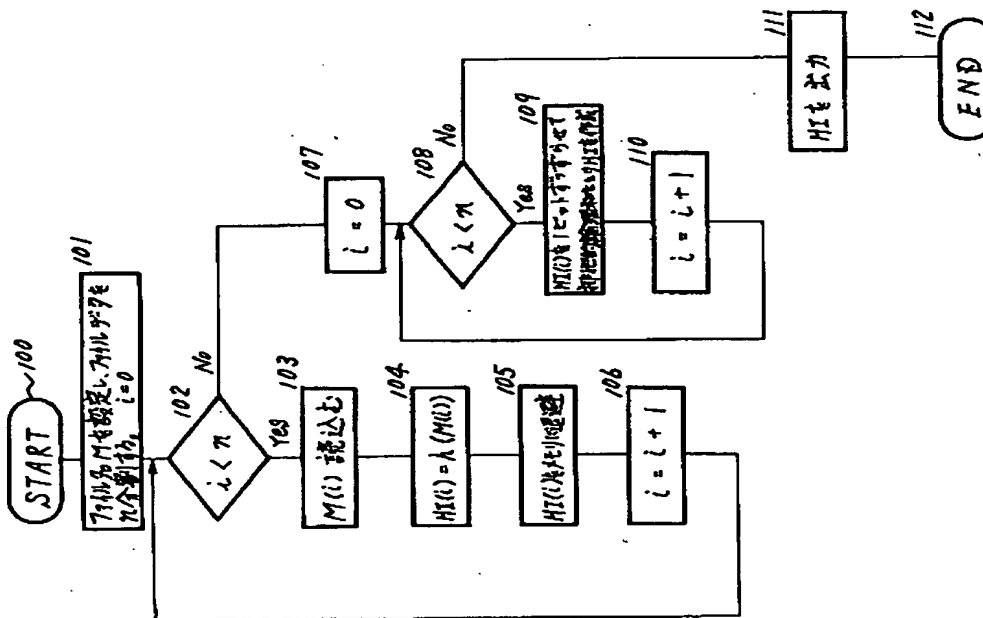
(21)

(7)

第 2 図

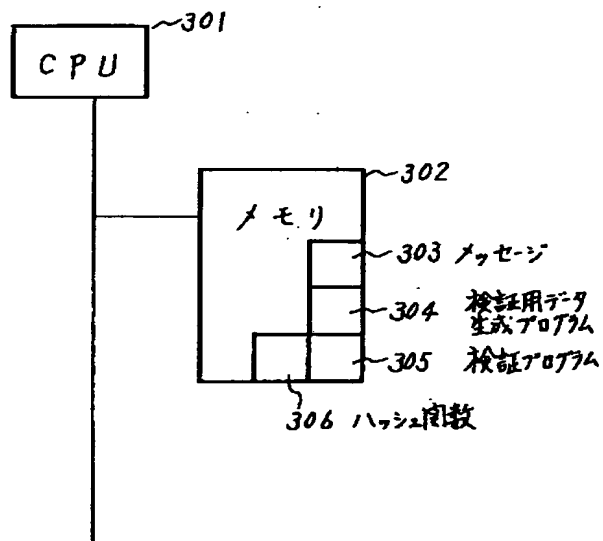


第 1 図

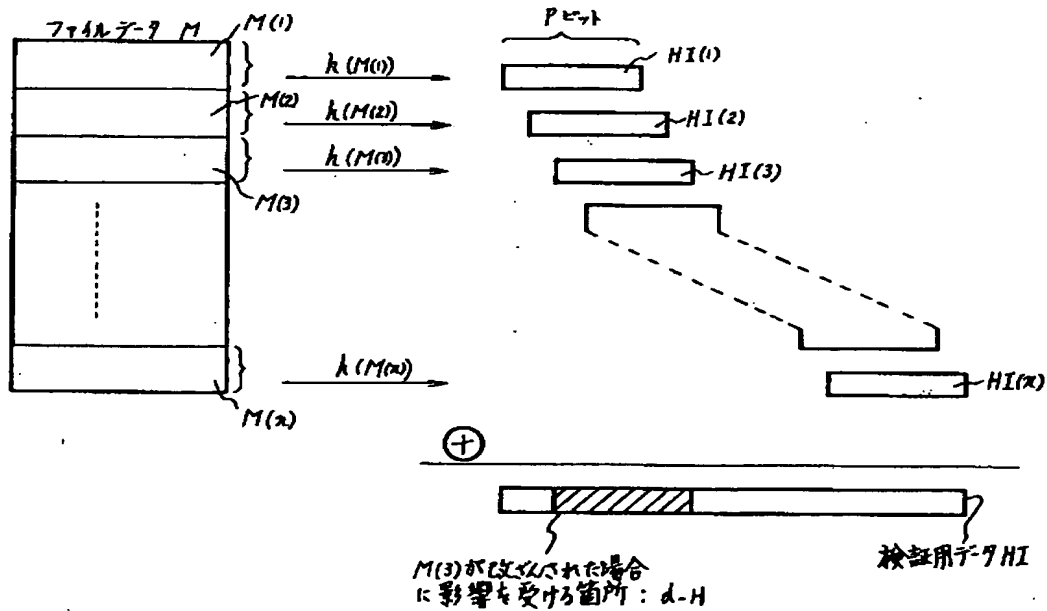


(8)

第 3 図

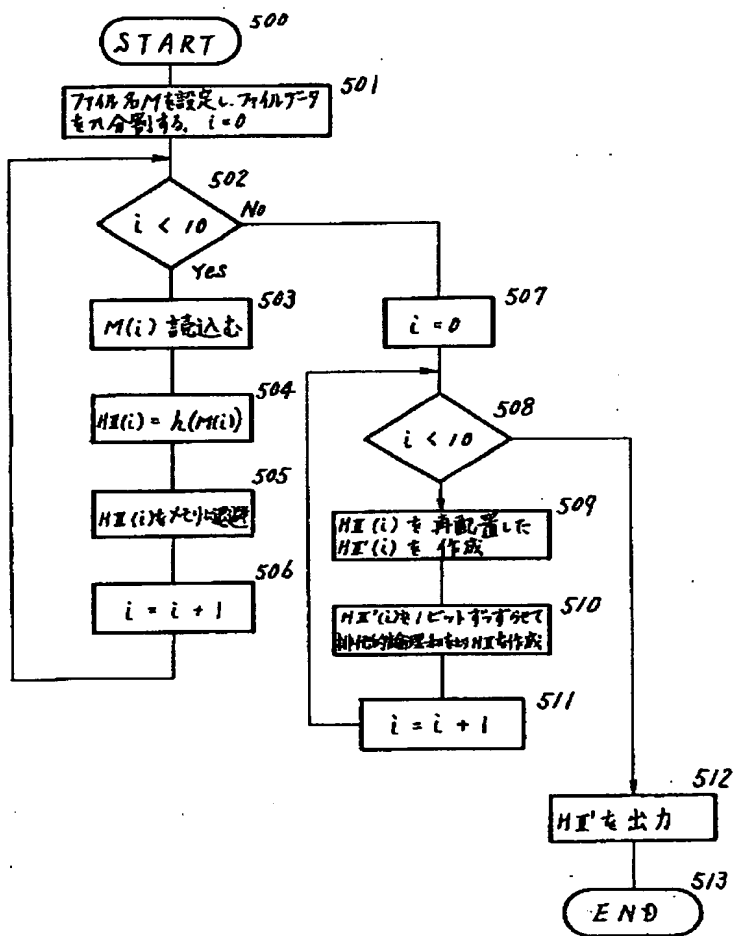


第 4 図

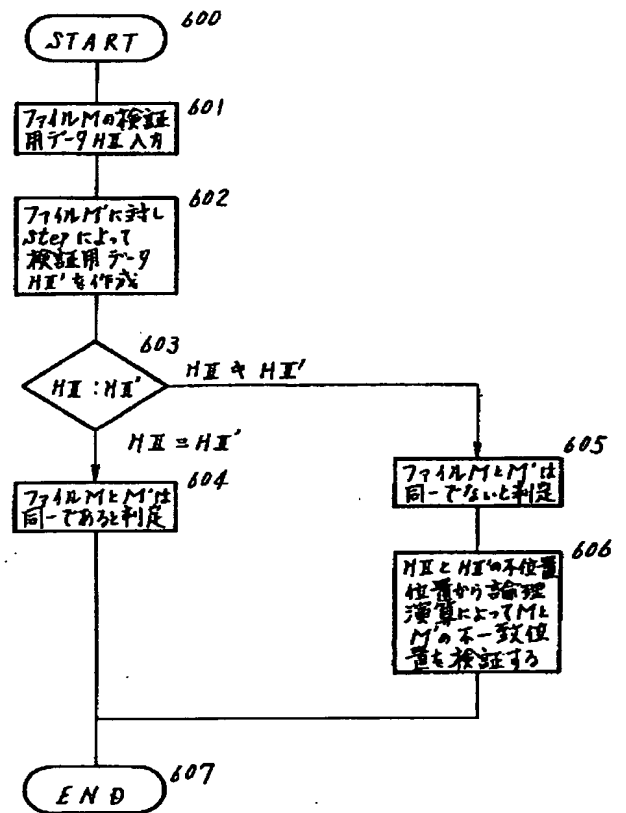


(9)

第 5 図



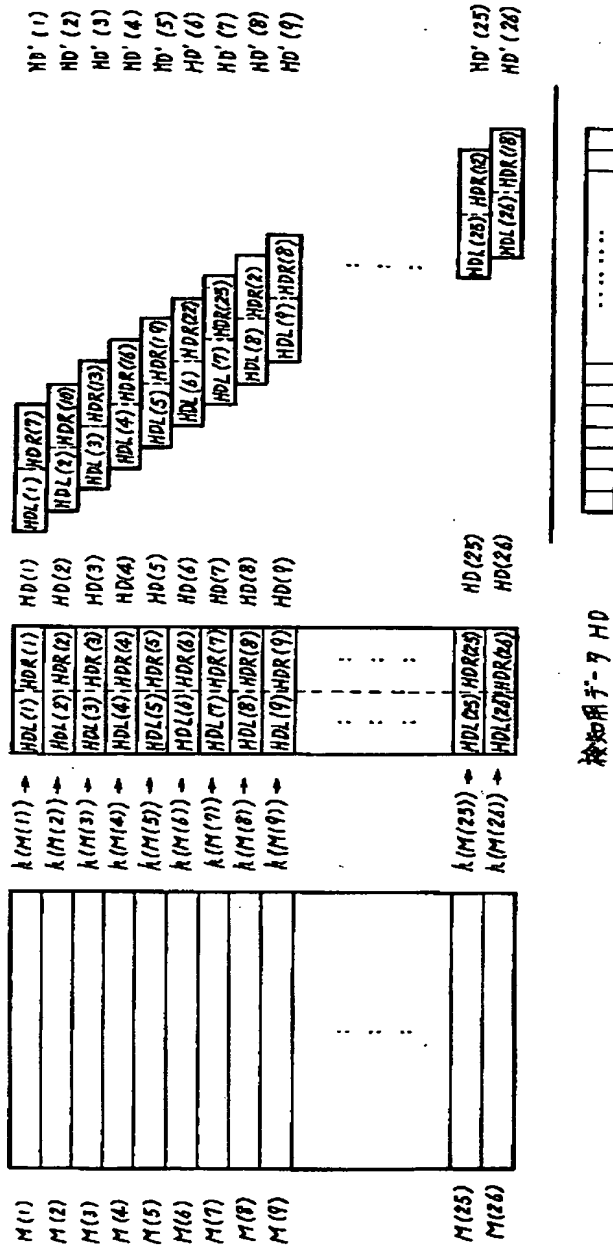
第 6 図



(10)

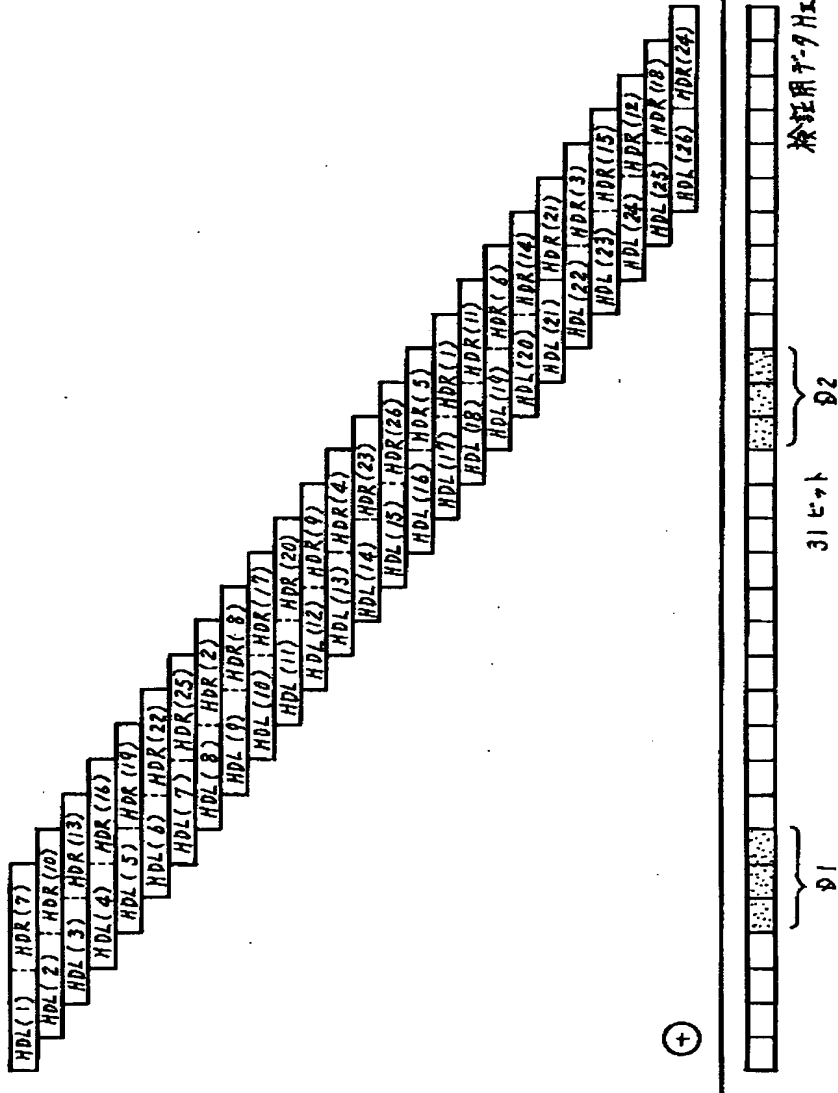
第 7 図

771N7-9M

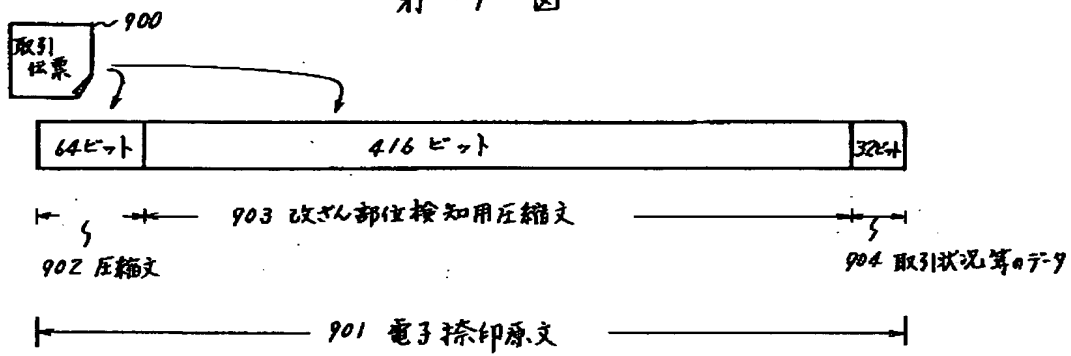


(11)

第 8 図



## 第 9 図



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**